

eRD20: Developing Simulation and Analysis Tools for the EIC



EIC SOFTWARE CONSORTIUM

Software Initiative for the EIC

EIC Software Consortium

Existing EIC Software

ANL TOPSiDE detector concept (ILC software variant)

BNL BeAST detector concept: EICroot (FairRoot variant)

BNL ePHENIX detector concept (fun4all)

JLAB JLEIC detector concept (GEMC → eJANA)

Software Review by EIC Community in November 2017

- **Actively maintained** ANL software, fun4All , EicRoot, then GEMC and now eJANA
- The analysis environments for the EIC will be chosen when the EIC experimental collaborations will form.
- Until then, we will examine the **requirements** for the EIC analysis environment and work on the **R&D** aspects of the EIC analysis environment.



EIC SOFTWARE CONSORTIUM

Goals and focus

- work on common interfaces among EIC simulation tools
- explore new avenues of software development (e.g., AI)
- **reach out to the EIC community**
 - communicate present status of EIC software
 - bring existing EIC software to end users
 - produce publicly available consensus-based documents on critical subjects
 - provide vision for the future

ESC members

ANL, BNL, JLAB, LUND,
INFN, SLAC, Trieste,
W&M (→ Manitoba)

Part of EIC Generic
Detector R&D program



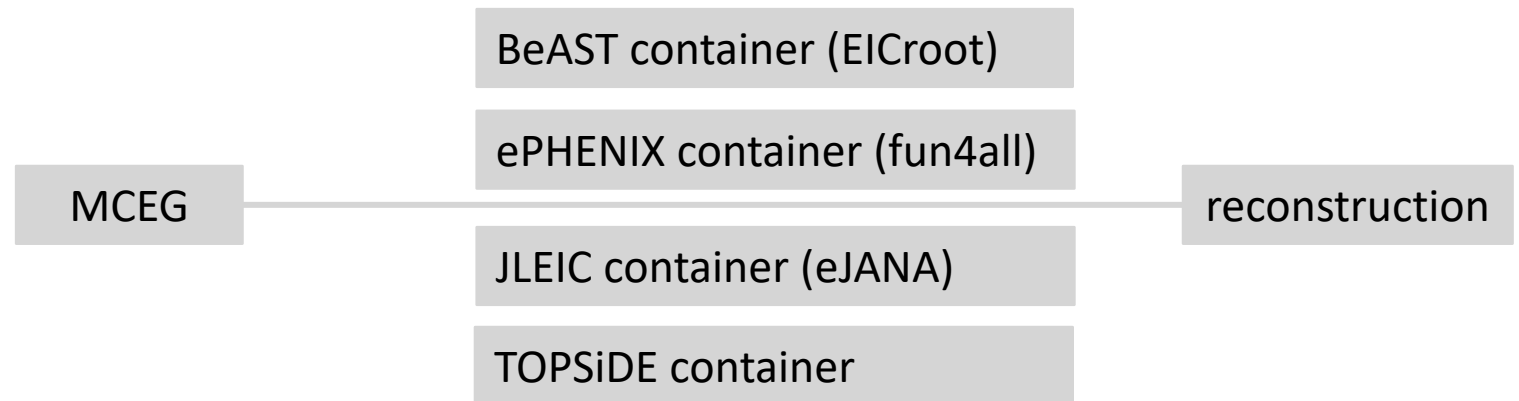
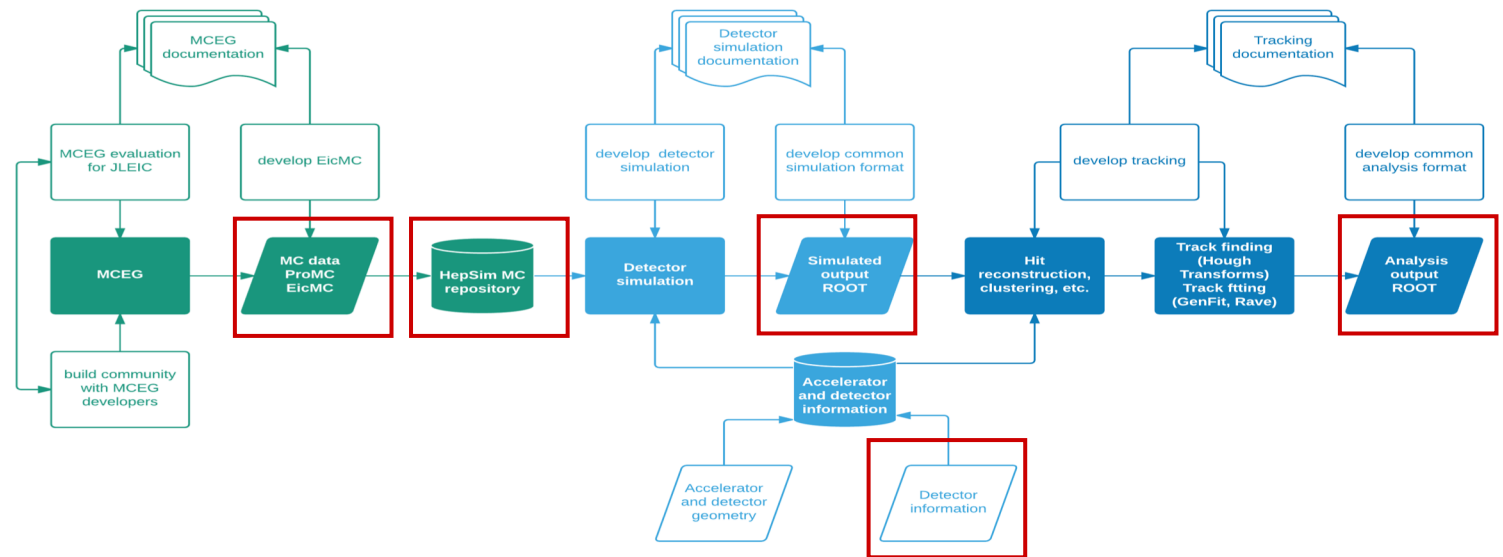
Common interfaces

Advice from ILC effort

- facilitate interoperability
- focus on exchange detector designs and data
 - get the event data model right and keep it open
 - pick a detector definition which is exchangeable

Norman Graf (SLAC)

"It's very difficult to herd cats keep physicists from re-inventing the wheel and writing new software packages."



ESC Initiatives



The poster features a vibrant background of glowing, intersecting lines in shades of blue, purple, and pink, creating a sense of depth and technology. At the top left, the EIC² and Jefferson Lab logos are displayed. The date 'TUESDAY, NOVEMBER 6' is prominently shown in a white box. The title 'MACHINE LEARNING SEMINAR' is centered in large, bold, black letters. Below the title, a detailed schedule of talks is listed, including topics like AI/ML algorithms, DOE Scientific Machine Learning, and neural network size requirements. The bottom of the poster includes the website www.jlab.org/indico/event/247.

EIC² Jefferson Lab

TUESDAY, NOVEMBER 6

MACHINE LEARNING SEMINAR

Machine Learning Seminar
CEBAF Center F113

11:00 **Opportunities for infusing physics and domain knowledge into AI/ML algorithms**
Prof. Animeshree Anandkumar (Caltech)

13:00 **DOE Scientific Machine Learning & AI Overview**
Dr. Steven Lee (DOE Advanced Scientific Computing Research)

13:30 **Study of neural network size requirements for approximating functions relevant to particle physics**
Jessica Stietzel (Notre Dame)

14:00 **NERSC's Machine Learning strategy**
Dr. Wahid Bhimji (NERSC)

14:30 **Discussion**

15:30 **Tag jet identification through the use of neural networks**
Anne-Katherine Burns (William & Mary)

15:50 **Machine intelligence applications for particle physics at Fermilab**
Dr. Aristeidis Tsaris (Fermilab)

16:30 **Overview of bayesian optimization applied to the GlueX case**
Cristiano Fanelli (MIT)

www.jlab.org/indico/event/247



The poster features a scenic night view of a city with illuminated buildings and a large church spire, reflected in a body of water. The title 'MCEGs' is written in large, bold, orange letters. Below it, the subtitle 'for future ep and eA facilities' is in blue. The date 'February 20-22, 2019' and location 'DESY Hamburg, Germany' are shown. The poster also mentions 'EIC User Group and MCnet present' and lists a program of updates, NLO simulations, GPDs/TMDs, and QED+QCD effects. Organizers are listed at the bottom, along with the website www.desy.de/mceg2019.

February 20-22, 2019
DESY Hamburg, Germany

EIC User Group and MCnet present

MCEGs
for future ep and eA facilities

PROGRAM

Updates to general-purpose MCEG for ep /eA
Status of NLO simulations for ep/eA
GPDs and TMDs in MCEGs
QED+QCD effects in ep/eA simulations

ORGANIZERS

Elke-Caroline Aschenauer (BNL) Simon Plätzer (University of Vienna)
Andrea Bressan (INFN Trieste) Stefan Prestel (Lund University)
Markus Diefenthaler (JLAB)
Hannes Jung (DESY)

www.desy.de/mceg2019



The poster features a night view of a city with illuminated buildings and a body of water. The title 'EIC Software Meeting' is written in large, bold, blue letters. Below it, the date 'May 20-21, 2019' and location 'Trieste, Italy' are shown. A text box describes the meeting's focus on simulation software status, tutorials, and contributions from the EIC Software Consortium and EICUG Software Working Group. It also mentions a joint session with the INFN School on 'Machine learning in High Energy Physics'. Organizers are listed at the bottom, along with the website <https://agenda.infn.it/event/17249/>.

EIC Software Meeting

May 20-21, 2019
Trieste, Italy

We will discuss the status of the simulation software for the EIC and will provide the tutorials for simulation tools. There will be contributions by members of the EIC Software Consortium and the EICUG Software Working Group as well as members from the HEP community. The meeting will also include a joint session with the INFN School on "Machine learning in High Energy Physics" that will be held in parallel to our meeting.

Organizers:
Andrea Bressan (INFN Trieste), Markus Diefenthaler (JLab), Alexander Kiselev (BNL)

For More Information:
<https://agenda.infn.it/event/17249/>

Engaging with the wider community

EICUG Software Working Group

EIC Software users and their requests

Ongoing EIC project

Software ✓
Documentation ✓
Requests none

Example projects

- ANL:TOPSiDE LDRD
- BNL: ePHENIX LoI
- BNL: eRHIC preCDR
- JLAB: JLEIC preCDR

Focus on (pre)CDRs and site selection as part of CD1 using existing lab software

EIC User Group

Common Software ✗
Common Documentation ✗
Requests software, documentation

Focus on preparation of EIC collaborations

- further develop EIC Science
- examine detector requirements
- work on detector designs
- work on detector concepts

requires simulations of physics processes and detector response

EIC Generic Detector R&D projects

Software ✓
Documentation ✗ - ✓
Requests common software

Request from Thomas Ullrich, manager of the R&D program:

Why we urgently need a

- **common (EIC-wide)**
- **easy to use***
- **capable**

detector simulation software

EIC Software Meeting on Detector and
Physics Simulations

Wednesday Jul 10, 2019, BNL

Thomas Ullrich (BNL)

* for a grad-student/postdoc with moderately good computing skills

Charge for EICUG Software Working Group

The EICUG Software Working Group's **initial focus** will be on **simulations of physics processes and detector response to enable quantitative assessment of measurement capabilities and their physics impact**. This will be pursued in a manner that is **accessible**, **consistent**, and **reproducible** to the EICUG as a whole.

It will embody simulations of all processes that make up the EIC science case as articulated in the white paper, eventually integrating new processes under request and with the help of interested communities within the EICUG. The Software working group is to engage with new major initiatives that aim to further develop the EIC science case, including for example the upcoming INT program(s), and is anticipated to play key roles also in the preparations for the EIC project(s) and its critical decisions. The working group will build on the considerable progress made within the EIC Software Consortium (eRD20) and other efforts. The evaluation or development of experiment-specific technologies, e.g. mass storage, clusters or other, are outside the initial scope of this working group until the actual experiment collaborations are formed.

The working group will be open to all members of the EICUG to work on EICUG related software tasks. It will communicate via a new mailing list and organize regular online and in-person meetings that enable broad and active participation from within the EICUG as a whole.

EIC Generic Detector R&D projects

Recommendation from January 2019

“The feedback from the user community indicates that there is a keen desire to have access to reliable and easy simulation and reconstruction frameworks. It seems that there remains a threshold for users to quickly engage in Monte Carlo simulations to carry out an end-to-end evaluation of various detector designs and study the EIC physics performance for different detector configurations.[...] The effort on providing common interfaces is strongly supported”

EIC Software Groups (beyond the simulation effort at the labs)

High Energy Physics

CERN ROOT

Possible collaboration

HEP Software Foundation

Started collaboration

MCnet

Started collaboration

SLAC Geant4

Established collaboration

Nuclear Physics

EIC Software Consortium

Community Endorsement ✗

Funding ✓

Same software suite Seamless data processing from DAQ to data analysis using AI

EIC Streaming Readout Consortium

Community Endorsement ✗

Funding ✓

EICUG Software Working Group

Community Endorsement ✓

Funding ✗

EIC Software initiative for community
Workflow environment for EICUG
to use (tools, documentation, support) **and**
to grow with user input (direction, documentation, tools)

EIC Software Meeting



EIC
Software
Meeting

May 20-21, 2019
Trieste, Italy

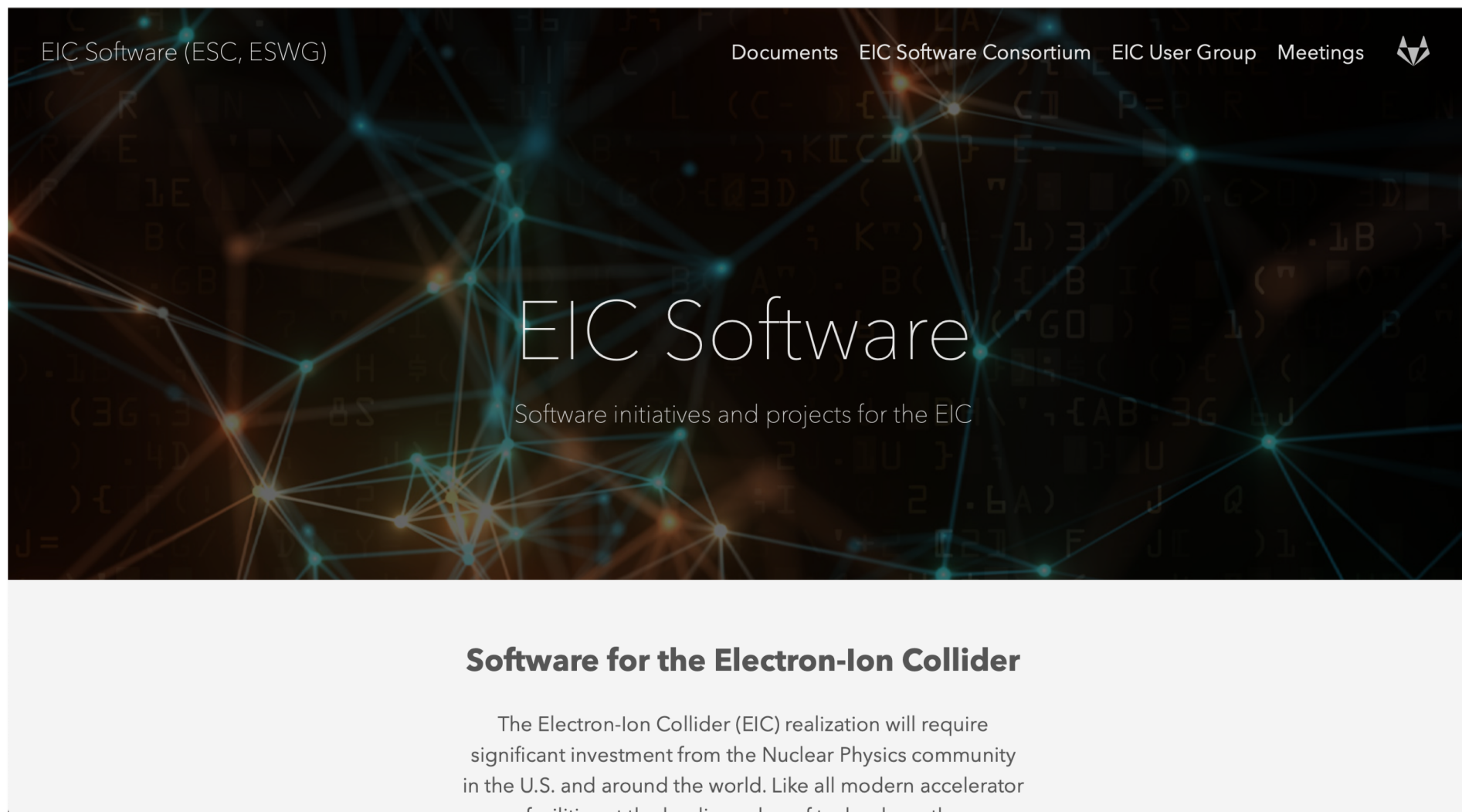
We will discuss the status of the simulation software for the EIC and will provide the tutorials for simulation tools. There will be contributions by members of the EIC Software Consortium and the EICUG Software Working Group as well as members from the HEP community. The meeting will also include a joint session with the INFN School on "Machine learning in High Energy Physics" that will be held in parallel to our meeting.

Organizers:
Andrea Bressan (INFN Trieste), Markus Dieffenthaler (JLab), Alexander Kiselev (BNL)

For More Information:
<https://agenda.infn.it/event/17249/>



Single point of entry



EIC Software website <https://eic.gitlab.io>

User interface for EIC simulations for EICUG

ejana-gui container image

JupyterLab web interface

ejana-app container image

eJana

Fast simulations (current focus)
Monte Carlo event generators
eic-smear used for EIC Whitepaper

TBD Alternative for fast simulations
Monte Carlo event generators
Geant4 in fast mode

Full simulations
Monte Carlo event generators
Geant4 (to-be-unified)
Event reconstruction toolkit

Detailed Status Reports

EICUG-MCnet: MCEG for future ep and eA facilities

Workshop summaries

Mar. 19 – 23 2018

POETIC-8

Feb. 20 – 22 2019

DESY

Elke-Caroline Aschenauer (BNL), Andrea Bressan (Trieste), Markus Diefenthaler (JLAB), Hannes Jung (DESY), and Simon Plätzer (Vienna)



www.desy.de/mceg2019

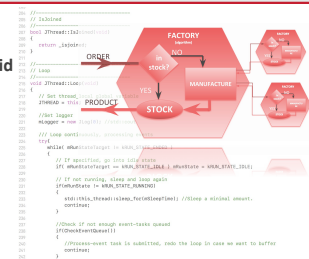
JANA2: Multi-threaded Event Reconstruction

Amber Boehnlein, Nathan Brei, David Lawrence, Dmitry Romanov
Jefferson Lab

May 21, 2019

EIC Software Meeting

Trieste, Italy



EIC Software Documentation

Wouter Deconinck, William & Mary

May 20-21, 2019
EIC Software Meeting



Supported by the National Science Foundation under Grant Nos. PHY-1405857, PHY-1714792



Kolja Kauder

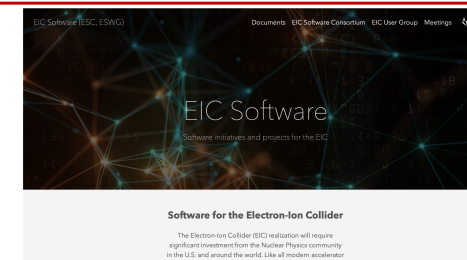
EIC Software Meeting
Trieste, May 2019

Users entry point



For community reference reconstruction and simulation

Single point of entry



EIC Software website <https://eic.gitlab.io>

EIC Software Meeting: Single point of entry

1

Jefferson Lab

simple

JupyterLab web interface

moderate

analysis scripts, python

complex

eJANA, plugins, C++

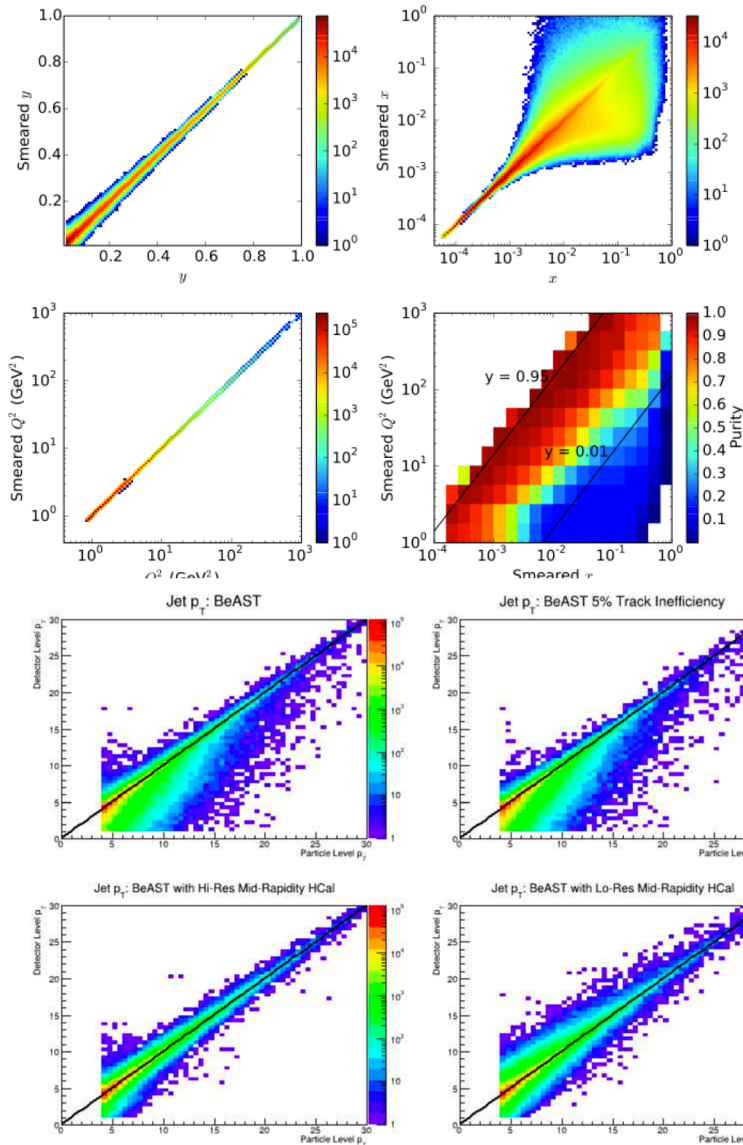
expert

JANA, eic-smear, ROOT, Geant4

Escaping complexity scaling trap

- provide interfaces to internal layers
- interaction between layers must be clear

Modularity each layer must be replaceable



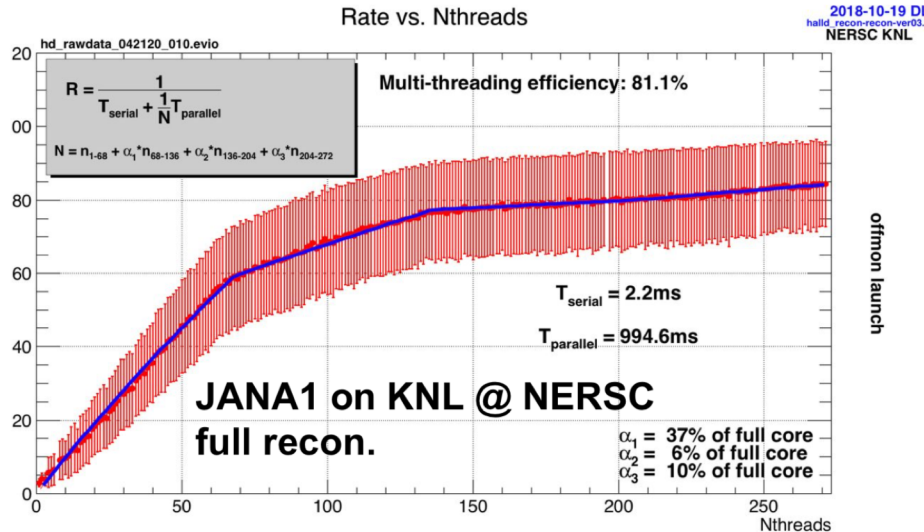
Fast simulations using ROOT, ideal for questions like

- “Given a (known) detector performance, how well can I measure some physics observable(s)?”
- “If I need to measure X with to some precision, what detector performance do I need?”
- Used extensively for **EIC White Paper**

Features

- interface to MCEGs for ep and eA
- smearing of overall detector performance:
 - can be easily modified in user code
 - includes acceptance effects
 - parametrizations for BeAST, ePHENIX, and ZEUS available
- ROOT trees for MC Truth and smeared information

EICUG Adapted as community tool for fast simulations



JANA C++ event processing framework

- **factory model** on demand interface, user-centered design
- multi-threaded with > 10 years experience
- **plugin support** provide mechanism for many physicists to contribute, multi-threading external to contributed code (parallelizer)

JANA2 under development (JLAB LDRD)

- take advantage of new C++ standards
- Python interface
- part of Streaming Readout Grand Challenge at Jefferson Lab (C++ streamed events processing framework)

JANA2

Multi-threaded HENP Event Reconstruction

[Welcome](#)
[Tutorial](#)
[Download + Docs](#)
[Installation + Use](#)
[FAQ](#)

// Welcome to JANA!

JANA is a C++ framework for multi-threaded HENP (High Energy and Nuclear Physics) event reconstruction. It is very efficient at multi-threading with a design that makes it easy for less experienced programmers to contribute pieces to the larger reconstruction project. The same JANA program can be used to easily do partial or full reconstruction, fully maximizing the available cores for the current job.

It's design strives to be extremely easy to setup when first getting started, yet have a depth of customization options that allow for more complicated reconstruction as your project grows. The intent is to make it easy to run on a laptop for local code development, but to also be highly efficient when deploying to large computing sites like [NERSC](#).

eJANA - stands for EIC JANA

- Basic reconstruction
- Physics analysis
- **Users detector codebase integration**

Reconstruction

- **Tracking** - Genfit
- **Vertex finding** – Rave
- **Physical analysis:**
 - ROOT C++ or
 - Python data science tools (Jupyter, Seaborn, Pandas, etc.)

Any existing C++ (or even others) code can be:

- compiled as JANA plugin
- run parallelized in eJANA
- accessed by other plugins



ESC summer students

Wouter Deconinck (W&M, Manitoba)

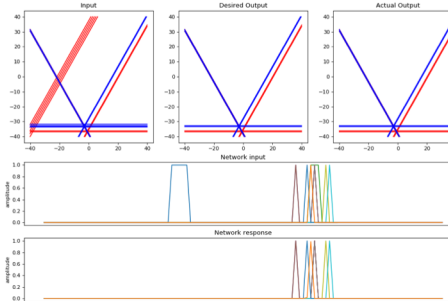


W&M Undergrad
Joseph Guy



W&M Undergrad
Nathan McConnell

Track finding in Jlab 12 data

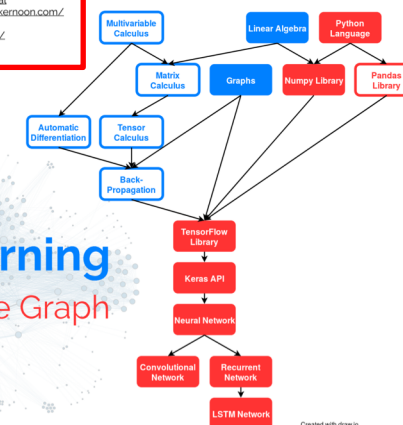


TensorFlow (TF)

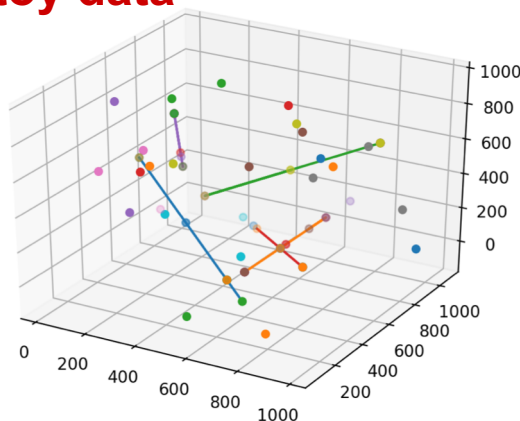
- Repository of Jupyter notebook TF tutorials for beginners: <https://github.com/4vassil/Labs/TF/tutorials>
- Repository of Jupyter notebook TF tutorials for beginners: <https://github.com/nlintz/TensorFlow-Tutorials>
- Written tutorial for beginners on the basics of TF: <https://www.datacamp.com/community/tutorials/tensorflow-tutorial>
- Written tutorial for beginners on the basics of TF: <https://hackernoon.com/machine-learning-with-tensorflow-8873dee2b68>
- Paper describing the internal workings of TF: <https://arxiv.org/pdf/1610.01178.pdf>

Machine Learning Knowledge Graph

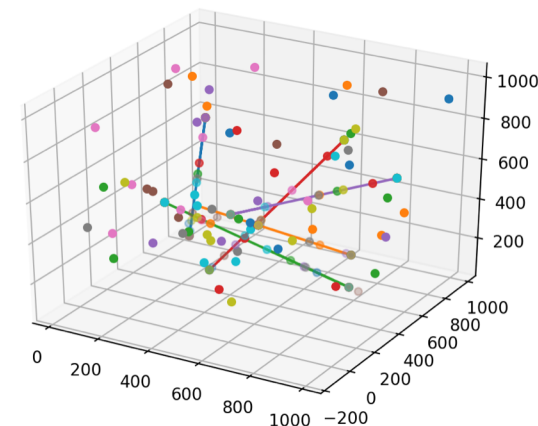
More topics: <https://ml-cheatsheet.readthedocs.io/en/latest/>



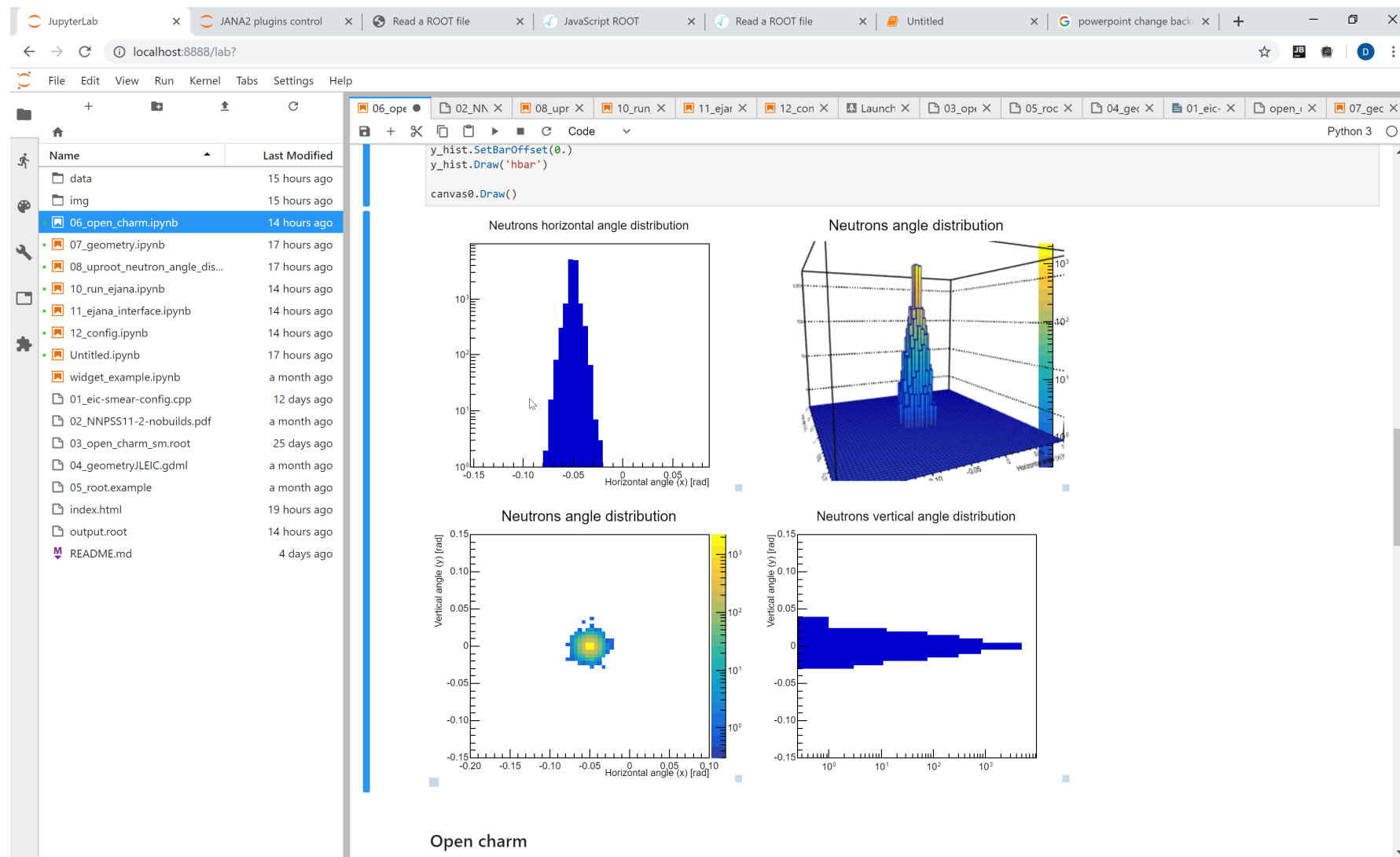
Track finding in toy data



using JLab JupyterHub at
<https://jupyter.jlab.org/>



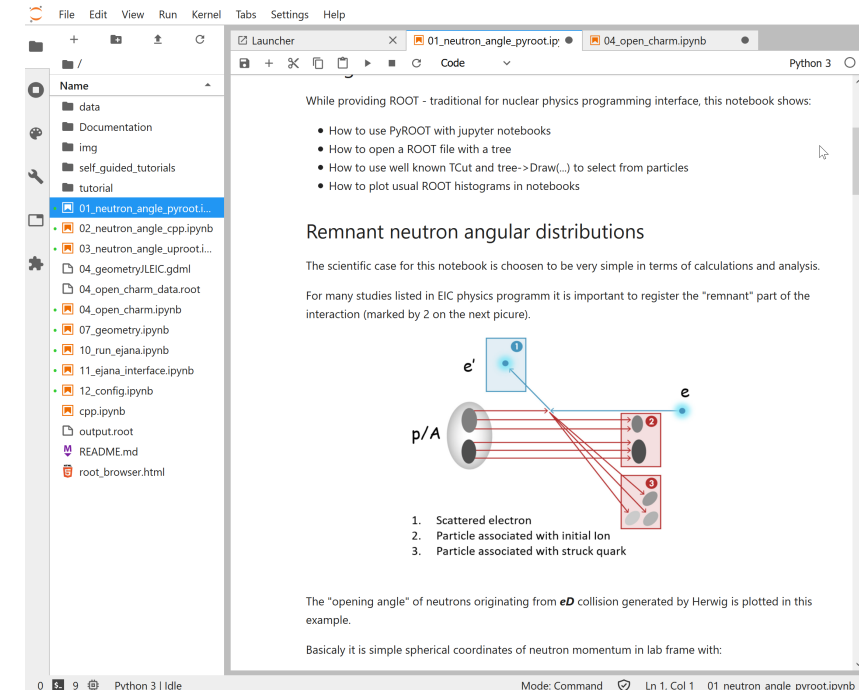
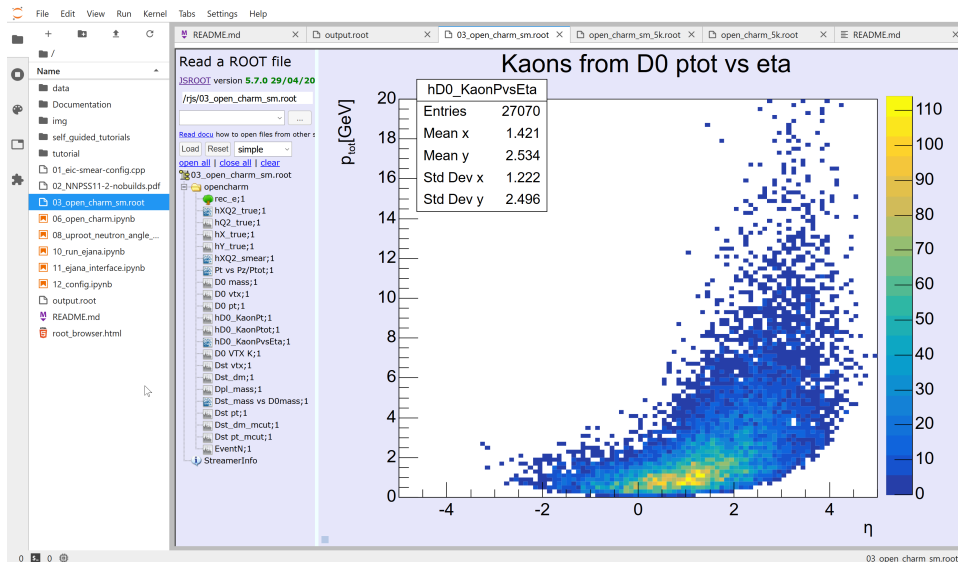
Workflow oriented interactive environment based on JupyterLab



Motivation for JupyterLab

Project Jupyter

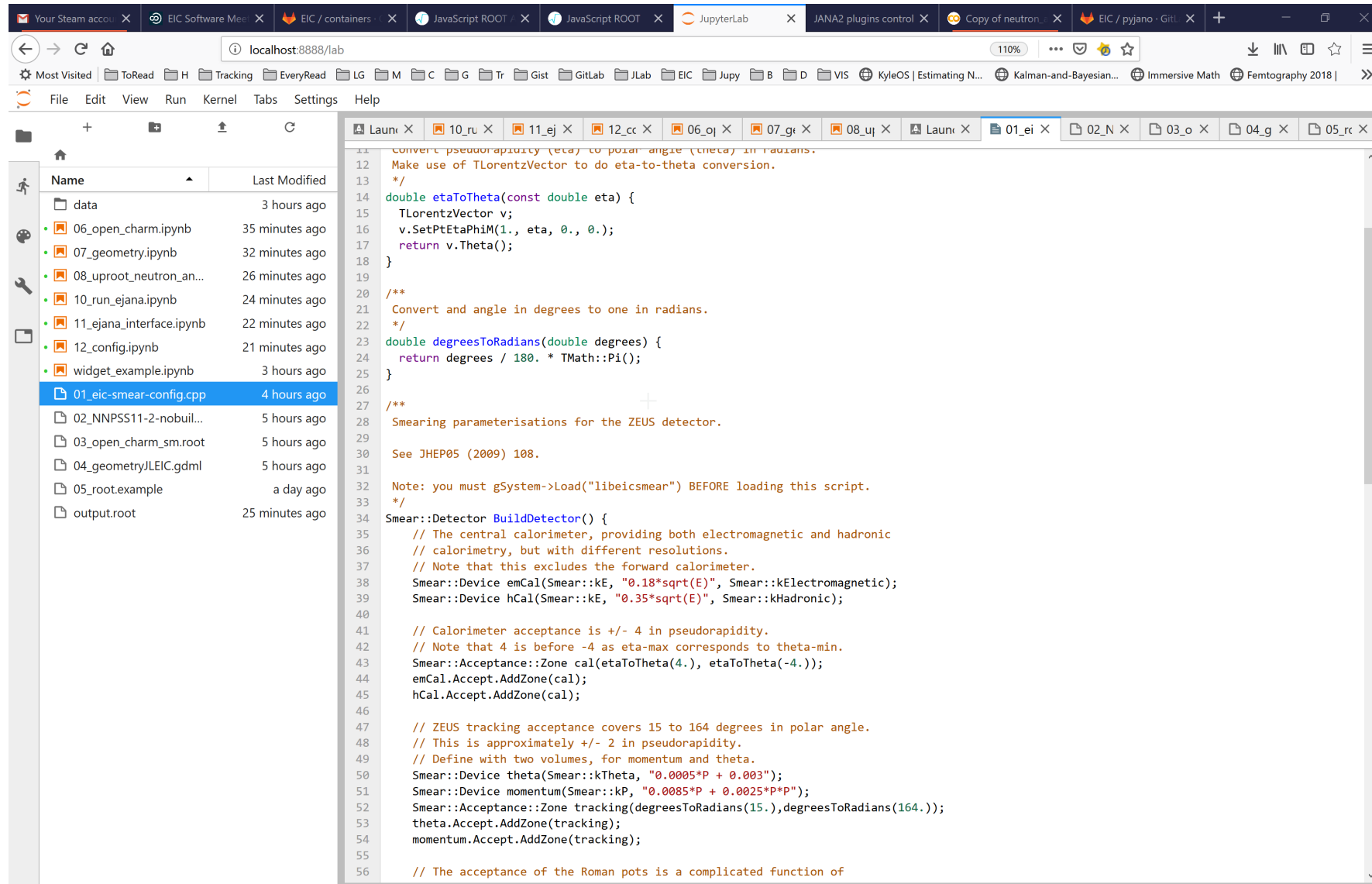
- bridge between modern Data Science and traditional Nuclear Physics methods
- self documenting



Ready to be

- cloud-based collaborative workspace
- medium for analysis, studies, reports (helping with reproducibility problem)

Complex layer: Interact with code



The screenshot displays a JupyterLab environment running on localhost:8888/lab. The interface includes a top toolbar with navigation and file management icons, a left sidebar with a file explorer, and a main area with a code editor. The file explorer shows a directory structure with files like 'data', '06_open_charm.ipynb', '07_geometry.ipynb', '08_uproot_neutron_an...', '10_run_ejana.ipynb', '11_ejana_interface.ipynb', '12_config.ipynb', 'widget_example.ipynb', '01_eic-smear-config.cpp', '02_NNPS11-2-nobuil...', '03_open_charm_sm.root', '04_geometryJLEIC.gdml', '05_root.example', and 'output.root'. The code editor shows the content of '01_eic-smear-config.cpp', which includes comments and C++ code for detector simulation. The code defines functions for converting pseudorapidity to polar angle and degrees to radians, and sets up a detector model with various parameters and acceptance regions.

```
11 Convert pseudorapidity (eta) to polar angle (theta) in radians.
12 Make use of TLorentzVector to do eta-to-theta conversion.
13 */
14 double etaToTheta(const double eta) {
15     TLorentzVector v;
16     v.SetPtEtaPhiM(1., eta, 0., 0.);
17     return v.Theta();
18 }
19
20 /**
21 Convert and angle in degrees to one in radians.
22 */
23 double degreesToRadians(double degrees) {
24     return degrees / 180. * TMath::Pi();
25 }
26
27 /**
28 Smearing parameterisations for the ZEUS detector.
29
30 See JHEP05 (2009) 108.
31
32 Note: you must gSystem->Load("libeicsmear") BEFORE loading this script.
33 */
34 Smear::Detector BuildDetector() {
35     // The central calorimeter, providing both electromagnetic and hadronic
36     // calorimetry, but with different resolutions.
37     // Note that this excludes the forward calorimeter.
38     Smear::Device emCal(Smear::kE, "0.18*sqrt(E)", Smear::kElectromagnetic);
39     Smear::Device hCal(Smear::kE, "0.35*sqrt(E)", Smear::kHadronic);
40
41     // Calorimeter acceptance is +/- 4 in pseudorapidity.
42     // Note that 4 is before -4 as eta-max corresponds to theta-min.
43     Smear::Acceptance::Zone cal(etaToTheta(4.), etaToTheta(-4.));
44     emCal.Accept.AddZone(cal);
45     hCal.Accept.AddZone(cal);
46
47     // ZEUS tracking acceptance covers 15 to 164 degrees in polar angle.
48     // This is approximately +/- 2 in pseudorapidity.
49     // Define with two volumes, for momentum and theta.
50     Smear::Device theta(Smear::kTheta, "0.0005*P + 0.003");
51     Smear::Device momentum(Smear::kP, "0.0085*P + 0.0025*P*P");
52     Smear::Acceptance::Zone tracking(degreesToRadians(15.), degreesToRadians(164.));
53     theta.Accept.AddZone(tracking);
54     momentum.Accept.AddZone(tracking);
55
56     // The acceptance of the Roman pots is a complicated function of
```

Moderate layer: Jupyter notebooks

The screenshot displays a JupyterLab environment running in a web browser at localhost:8888/lab. The interface includes a file browser on the left, a code editor in the center, and a console/output area at the bottom.

File Browser: Shows a directory structure with files like `data`, `06_open_chem.ipynb`, `07_geometry.ipynb`, `08_uproot_neutron_an...` (selected), `10_run_ejana.ipynb`, `11_ejana_interface.ipynb`, `12_config.ipynb`, `widget_example.ipynb`, `01_eic-smear-config.cpp`, `02_NNPSS11-2-nobuil...`, `03_open_chem_sm.root`, `04_geometry/JEIC.gdml`, `05_root.example`, and `output.root`.

Code Editor: Contains the following Python code:

```
[3]: tree = uproot.open('./data/eventless_output.root')['eventless']['tree']
tracks_df = tree.pandas.df(["pdg_b", "p_b", "px_b", "py_b", "pz_b"])

[10]: tracks_df['vertical_angle'] = np.arcsin(tracks_df['py_b']/tracks_df['p_b'])
tracks_df['horizontal_angle'] = np.arctan(tracks_df['px_b']/tracks_df['pz_b'])
v_angle = tracks_df[tracks_df.pdg_b == 2212].vertical_angle.values
h_angle = tracks_df[tracks_df.pdg_b == 2212].horizontal_angle.values

Horizontal angle (x):  $\alpha_h = \arctan\left(\frac{p_x}{p_z}\right)$ 

Vertical angle (y):  $\alpha_v = \arcsin\left(\frac{p_y}{p}\right)$ 

[11]: fig, ax = plt.subplots()
h, xedges, yedges, im = ax.hist2d(h_angle, v_angle,
bins=[np.arange(-200,101,5)/1000,np.arange(-150,151,5)/1000],
norm=matplotlib.colors.LogNorm())
plt.colorbar(im, ax=ax)
plt.xlabel('Horizontal angle (x) [rad]')
plt.ylabel('Vertical angle (y) [rad]')
plt.title('Neutrons angle distribution')
ax.set_axisbelow(True)
plt.grid(True)
plt.show()
```

Figure: A 2D histogram titled "Neutrons angle distribution". The x-axis is labeled "Horizontal angle (x) [rad]" and ranges from -0.20 to 0.10. The y-axis is labeled "Vertical angle (y) [rad]" and ranges from -0.15 to 0.15. The plot shows a dense cluster of data points centered around (0,0), with a color bar on the right indicating a logarithmic scale from 10^0 to 10^3 .

Simple layer: Interactive widgets

The screenshot displays a JupyterLab environment running in a web browser at localhost:8888/lab. The interface includes a file browser on the left, a code editor in the center, and a console at the bottom. The file browser shows a list of files, with '12_config.ipynb' selected. The code editor shows two code cells. The first cell imports the 'pyjano' package and creates a 'Jana' object. The second cell calls 'jana.plugins_gui()', which displays a GUI with two columns of toggle switches. The left column, titled 'IO plugins:', contains 'lund_reader', 'beagle_reader' (which is checked), 'hepmc_reader', 'jleic_geant_reader', and 'jleic_gemc_reader'. The right column, titled 'Process & Analysis:', contains 'trk_fit', 'trk_eff', 'jleic_iff', 'jleic_occupancy', 'vmeson', and 'open_charm' (which is checked). A 'verbose (int)' control is set to 2. Below the GUI, a text block describes the 'beagle_reader' plugin, stating it opens files from the BeAGLE event generator as a data source and provides a link to its documentation.

```
[6]: from pyjano import Jana
jana = Jana()

[7]: jana.plugins_gui()
```

IO plugins:

- ☐ lund_reader
- ☒ beagle_reader
- ☐ hepmc_reader
- ☐ jleic_geant_reader
- ☐ jleic_gemc_reader

Process & Analysis:


- ☐ trk_fit
- ☐ trk_eff
- ☐ jleic_iff
- ☐ jleic_occupancy
- ☐ vmeson
- ☒ open_charm

verbose (int): ☒ 2

Plugin **beagle_reader**: Opens files from BeAGLE event generator as a data source
BeAGLE - Benchmark eA Generator for LEptoproduction
[Documentation](#)

EIC Software Meeting on Detector and Physics Simulations

EIC Software Meeting on Detector and Physics Simulations

 Wednesday 10 Jul 2019, 09:00 → 15:00 US/Eastern

 2-160 (BNL Physics)

 Andrea Bressan (Trieste) , Markus Diefenthaler (Jefferson Lab) , Torre Wenaus (BNL)

35 participants

Description The [Electron-Ion Collider User Group](#) (EICUG) is organizing an EIC Ad-hoc Meeting on detector and physics simulations:

We will work out the requirements for simulation tools including geometry exchange between the eRHIC and JLEIC concepts and plan next steps for common tools for the EICUG. The work will be also relevant for TOPSiDE and other detector concepts for the EIC.

The goal of the meeting is to clarify the strategy for EIC detector and physics simulations and to meet on the time scale of FY19/FY20 the requirements for common tools and documentation in the EICUG. We would like to include all EIC Software initiatives in our discussion.

The meeting will be held at BNL on July 10, prior to the EIC Generic Detector R&D meeting on July 11-12, and will include presentations from HEP and NP experts as well as discussions with EIC Software developers.

The meeting has a hard finish at 3pm (we lose the room).

If you wish to attend in person and do not have a current BNL badge, please contact Torre Wenaus (wenaus@gmail.com, 631-681-7892) as soon as possible. Also contact Torre with any BNL related issues in general.

BlueJeans: <https://bluejeans.com/920347364>

Detector R&D and Simulations (Case 1)

- For almost all proposed detector R&D the Committee has and does request extensive simulations showing the concept is sound before substantial money is released for hardware/prototyping/test beams etc.
 - Geant simulations are mature and sophisticated enough to allow one to test the validity of the proposed concept and establish the detector performance to good accuracy
 - In most cases only GEANT simulations are needed
 - no need for sophisticated framework
 - no need for elaborate tracking software
 - The key issue in all cases is the detector description
 - geometry
 - materials

3

Why “easy accessible and usable”?

- Many R&D groups are also involved in other projects that have a mature simulation setup (e.g. CMS, PANDA, COMPASS)
- If the EIC does not provide a common and easy simulation setup the chances are high groups will either use what they are used to or even go to vanilla Geant4
- This makes the situation even worse
- Note that other groups (e.g. ILC) have solved this successfully. Changing one detector to another is simply a matter of one switch. This makes it easy for groups working on certain sub-detectors to see how it fits and performs in whatever setup is available

8

Detector R&D and Simulations (Case 2)

- Optimization
 - No detector concept is perfect from the start
 - Multi-parameter problems
 - Example Si-vertex detectors: # disks/layers, position of disks/layers, pixel size etc.
 - Tracking/RICH: composition of gases, voltages, readout layers (GEM), etc
 - It is impossible to build and test prototypes for all possible configurations - phase space is too big. Often simulations are the only way to find the right parameter set
- Simulations are mandatory here

4

Keep Context in Mind

- Users within the R&D community are **not** looking for a full flashed-out framework at this time
- A simple *lite setup* with a well defined geometry description “standard” might get them a long way as long if it is EIC wide and easy to use
- It is understood that a complete geometry/material package has to fulfill many tasks: simulations, reconstruction, all with condition DB interface, but this is something the collaborations will have to work out later. If the EIC User Group finds a workable solution now, chances are high that it will be picked up by the actual collaborations later.

9

- agree on detector naming conventions
- agree on G4 C++ as the basis of detector description for now (access to all G4 volumes including new recommended ones like tessellated solids)
 - 3 of the 4 approaches (not EicRoot) can work with C++ G4 detector descriptions
 - make the (inevitable) framework-specific adaptation layer needed around the C++ as thin as possible
- agree on how structs for parameters are defined and managed
 - 'numbers in the code' is not necessarily a bad thing if they are managed carefully. Coming to commonality on a more long term solution like DB is a bigger issue (and not on Thomas' 'now' list)
- agree on API/class design for sensitive detector stepping action/digi
- agree on hits output structure
 - hits structure - can we agree on that? depends on detector and how reco is handled afterwards
 - even without this we've agreed on enough to satisfy Thomas
- common repository we point people to when they come along with a detector model to plug in
- deliverable: the simple document and template(s) we give to the user on how to implement and integrate their new subdetector in the detector concepts via our common Geant4 infrastructure
 - time frame?
 - by end of the fiscal year (end Sep), have made the agreements concrete in a document
 - for implementation...?
- deliverable: an updated/agreed document of the requirements
- need follow-up meeting on participation, who will do what by when
 - what resources do we have and how do we use them

Meeting schedule

Date	Topic(s)
05/20 – 05/22	EIC Software Meeting in Trieste, Italy
06/27	Benchmarks and validation (only remote)
07/10	EIC Software Meeting on detector and physics simulations
07/23	Tutorial during EICUG Meeting in Paris, France
08/	Summer break
09/24	Geant4 Technical Forum on EIC
to be annouced	

- Geant4 Technical Forum will be held at Jefferson Lab on Tuesday, September 24 2019, as a part of the Geant4 Collaboration Meeting.
 - 16:00-18:00 US Eastern time (22:00-24:00 Central European time)
- This Technical Forum is highlighting EIC:
 - Geant4 developers will update the EIC community of the recent and ongoing developments that are relevant for EIC,
 - Requirements and concerns from the EIC community should be presented,
 - and discussed.
- Please keep Makoto and me informed on
 - What you want to hear from Geant4 developers, and
 - What you want to tell / argue to Geant4 developers

FY20 budget request

Funds for hiring a postdoctoral researcher of USD 120,000

- works solely on the community tools for physics and detector simulations including geometry exchange between eRHIC and JLEIC and BeAST, ePHENIX, JLEIC, TOPSiDE detector concepts
- service for the EIC and in particular Detector R&D communities: support for using the tools and for integrating existing detector designs and reconstruction algorithms into the community tools

Travel funds of USD 20,000

- in-person meetings to work together on key tasks
- invite scientists that are essential to the R&D effort

Budget scenarios

- Nominal budget (100%): USD 140,000
- Nominal budget (-80%): USD 20,000 (only travel)

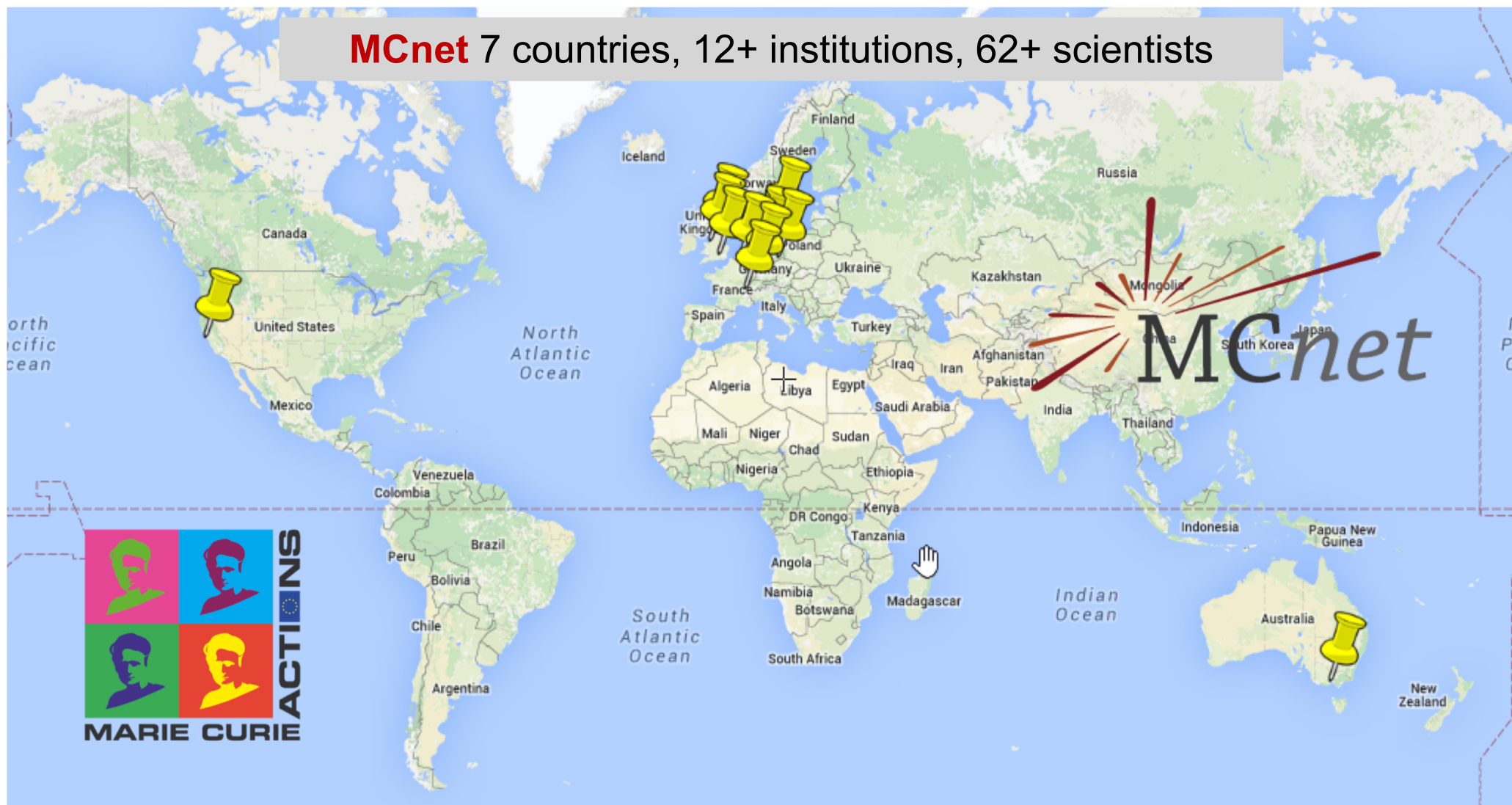


EIC SOFTWARE CONSORTIUM

Additional slides

MCEG Initiative

MCEG Developers



Workshops: MCEGs for future ep and eA facilities



February 20-22, 2019
DESY Hamburg, Germany

EIC User Group and MCnet present

MCEGs

for future ep and eA facilities

PROGRAM

Updates to general-purpose MCEG for ep /eA
Status of NLO simulations for ep/eA
GPDs and TMDs in MCEGs
QED+QCD effects in ep/eA simulations

ORGANIZERS

Elke-Caroline Aschenauer (BNL) Simon Plätzer (University of Vienna)
Andrea Bressan (INFN Trieste) Stefan Prestel (Lund University)
Markus Diefenthaler (JLAB)
Hannes Jung (DESY)

www.desy.de/mceg2019

MCEG2018 19–23 March 2018

- Started as satellite workshop during POETIC-8

P O E T I C 8

8th International Conference on Physics Opportunities at an Electron-Ion Collider

19-23 March 2018, University of Regensburg

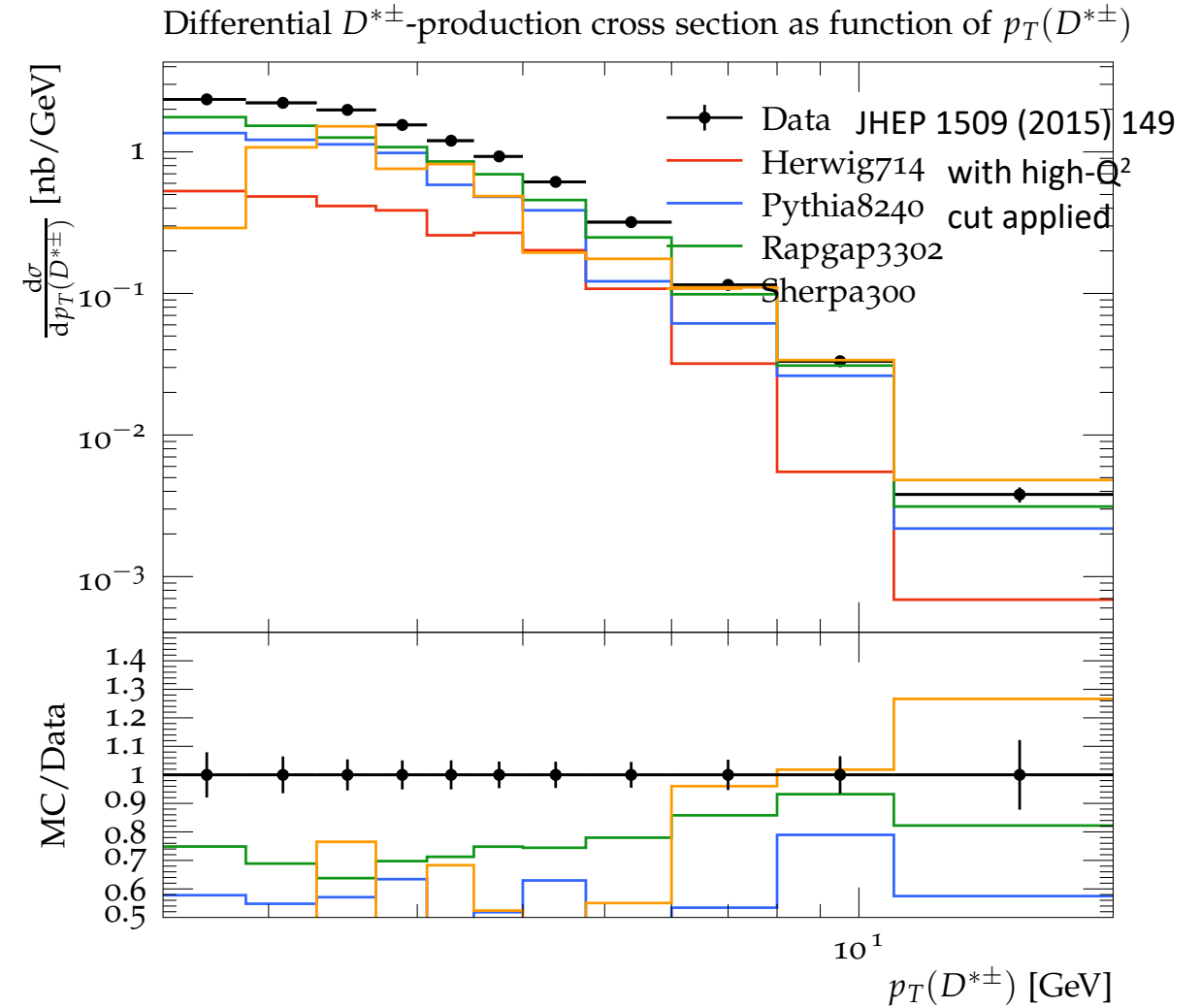
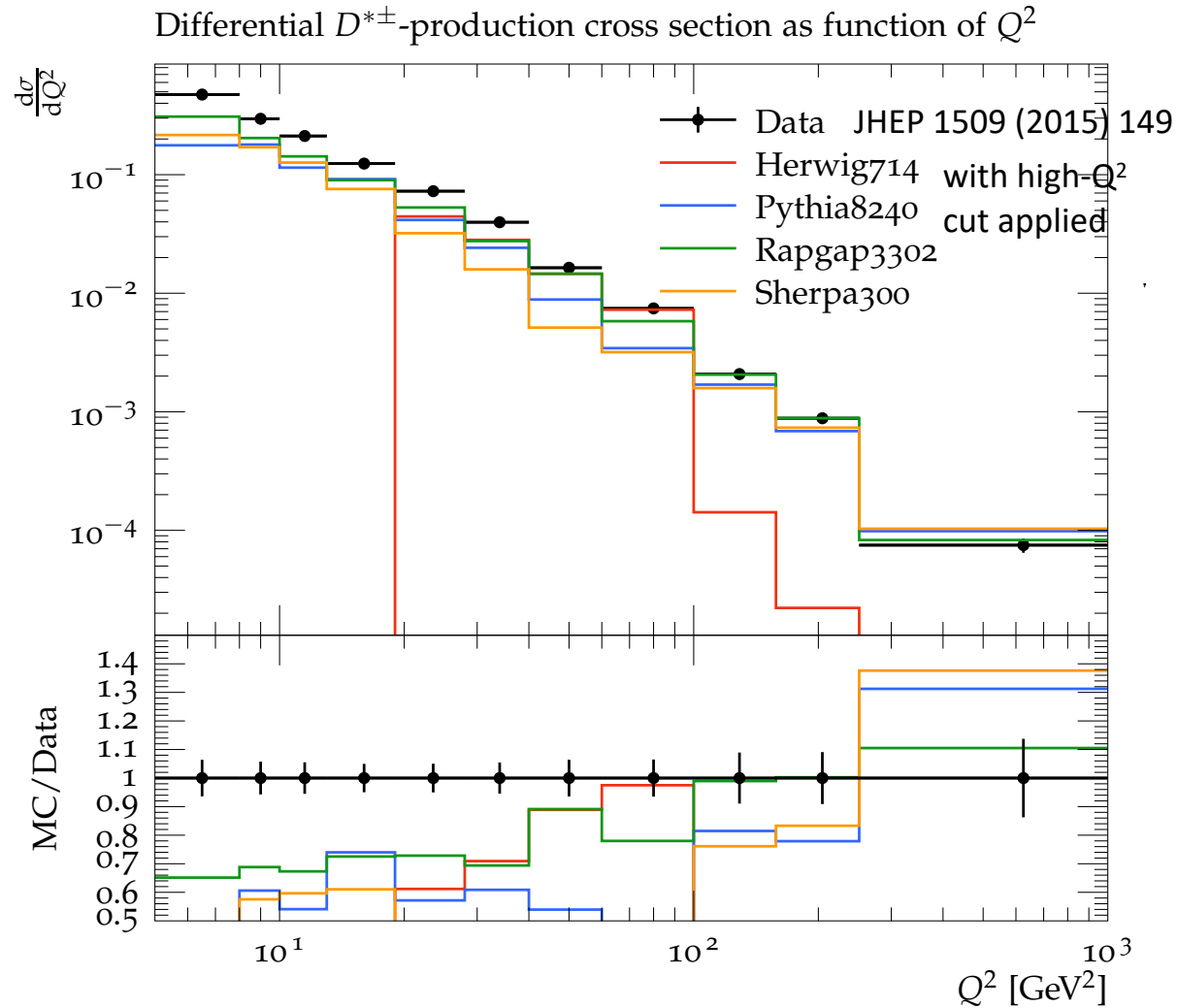
- Collaboration EICUG-MCnet

Goal of workshop series

- Requirements for MCEGs for ep and eA
- R&D for MCEGs for ep and eA

MCEG2019 20–22 February 2019

- Status of ep and eA in general-purpose MCEG
- Status of NLO simulations for ep
- TMDs and GPDs and MCEGs
- Merging QED and QCD effects



Container for Pythia8+DIRE by Nadine Fischer (Pythia)

Jupyter README 8 minutes ago Logout

File Edit View Language Plain Text

```
1 Welcome to the Jupyter notebooks for Pythia 8 and DIRE!
2
3
4 You have the choice to run the following notebooks:
5
6 pythiaPI.ipynb
7 Gives a basic idea of the Pythia 8 event generator, by using the Python
8 interface of Pythia 8. You can adjust a set of parameters and choose
9 from different different histograms to be plotted.
10
11 pythiaRivetPI.ipynb
12 Shows how to use the Pythia 8 event generator, together with Rivet,
13 by using the Python interface of Pythia 8.
14
15 pythiaRivet.ipynb
16 Shows how to use Pythia 8, together with Rivet, by using an already
17 compiled executable called pythiaHepMC. You can adjust a set of parameters
18 and a settings file is created.
19
20 pythiaRivetUS.ipynb
21 As pythiaRivet.ipynb, but uses a prepared settings file, to be provided
22 by the user.
23
24 direRivet.ipynb
25 Shows how to use Pythia 8 with the DIRE parton shower, together with
26 Rivet, by using the default DIRE executable. You can adjust a set of
27 parameters and a settings file is created.
28
29 direRivetUS.ipynb
30 As direRivet.ipynb, but uses a prepared settings file, to be provided
31 by the user.
32
33 direEvent.ipynb
34 Pythia 8 with the DIRE parton shower, graphical output of one event
35 with the default DIRE executable.
36 The process can be chosen as well as a few basic parameters.
37
38 tuning.ipynb
39 Tuning with Professor, Rivet, and Pythia 8 / DIRE.
40
```

Jupyter notebook interface

Pythia 8 standalone

This notebook gives a basic idea of the Pythia 8 event generator, by using the Python interface of Pythia 8. You can adjust a set of parameters and choose from different different histograms to be plotted.

First, lets import all necessary modules.

```
In [1]: import os, sys, pythia8
from plotting import MULTHIST
import py8settings as py8s
```

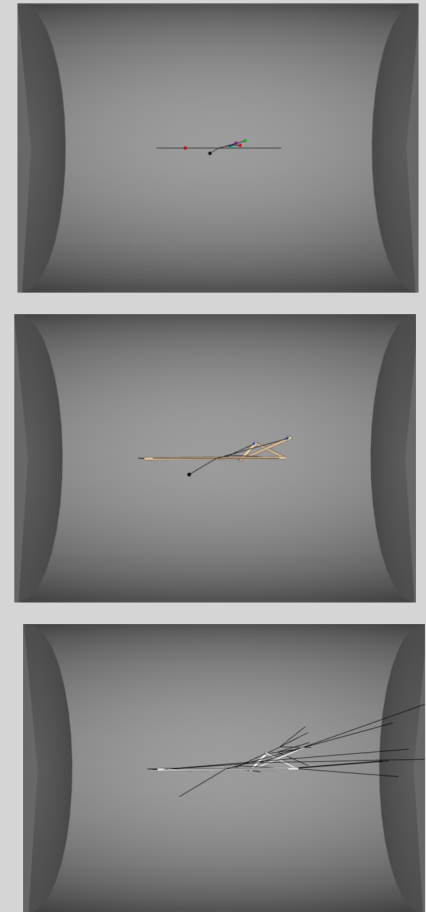
Now we create a Pythia 8 object and apply the settings to define the incoming beams. More settings can be adjusted later.

```
In [2]: # Setup pythia, apply beam settings.
pythia = pythia8.Pythia()
py8s.beam_settings(pythia)
```

You can now set the parameters for the incoming beams:

beam A id [Beams:idA]	e-
beam B id [Beams:idB]	p
beam frame type [Beams:frameType]	2: back-to-back beams with different energies, set Beams:eA and Beams:eB
CMS energy for Beams:frameType = 1 [Beams:eCM]	65.7
beam A energy for Beams:frameType = 2 [Beams:eA]	10.8
beam B energy for Beams:frameType = 2 [Beams:eB]	100

Visualization of ep collision



MCEG–HERA comparisons and MCEG validation for ep

Rivet example

SIDIS analysis at HERMES

```
66 // Extract the particles other than the lepton
67 const FinalState& fs = apply<FinalState>(event, "FS");
68 Particles particles;
69 particles.reserve(fs.particles().size());
70 const GenParticle* dilepGP = dl.out().genParticle();
71 foreach (const Particle& p, fs.particles()) {
72     const GenParticle* loopGP = p.genParticle();
73     if (loopGP == dilepGP)
74         continue;
75     particles.push_back(p);
76 }
77
78 // Apply HERMES cuts.
79 bool validx = (x > 0.023 && x < 0.6);
80 if (q2 < 1. || w2 < 10. || y < 0.1 || y > 0.85 || !validx)
81     vetoEvent;
82
83 // good inclusive event, let's do bookkeeping before we look at the hadrons
84 dis_tot += weight;
85 dis_x->fill(x, weight);
86 dis_Q2->fill(q2, weight);
87
88 for (size_t ip1 = 0; ip1 < particles.size(); ++ip1) {
89     const Particle& p = particles[ip1];
90
91     // get the particle index, check if it is a particle of interest
92     const int part_idx = get_index(p.genParticle()->pdg_id());
93     if (part_idx < 0) {
94         continue;
95     }
96
97     // we have a particle of interest, let's calculate the kinematics
98     // z
99     const double z = (p.momentum() * pProton) / (pProton * q);
100     // pt
101     const double pth = sqrt(p.momentum().pT2());
102
103     // get our z index, if negative, we have a particle outside of [.2, .8]
104     const int z_idx = calc_zslice(z);
105     if (z_idx < 0) {
106         continue;
107     }
108
109     // store the events and make cuts where necessary
110     //
111     // pt cut for variables not binned in pt
112     if (pth > 0 && pth < 1.2) {
113         mult_z[part_idx]->fill(z, weight);
114         mult_zx[part_idx][z_idx]->fill(x, weight);
115         mult_zQ2[part_idx][z_idx]->fill(q2, weight);
116     }
117     mult_zpt[part_idx][z_idx]->fill(pth, weight);
118 }
```

- **MCEG R&D** requires *easy access to data*
- data := analysis description + data points
- **HEP** existing workflow for MCEG R&D using tools such as Rivet and Professor
- **Detailed comparisons between modern MCEG and HERA data**
 - workshop on [Rivet for ep](#) (Feb 18—20 2019)
 - mailing list rivet-ep-l@lists.bnl.gov
 - HERA data not (yet) included in MCEG tunes

Merging QED and QCD effects

CLASSIFICATION OF $O(\alpha)$ QED CORRECTIONS

- **Radiation from the lepton**
model independent (universal),
dominating by far: enhanced by large logs, $\ln(Q^2/m_e^2)$
- vacuum polarization (boson self energy)
universal, photon self energy $\rightarrow \alpha_{em}(Q^2)$
- **Radiation from the hadronic initial/final state**
parton model: radiation from quarks
to be considered as a part of the nucleon structure
- **Interference of leptonic and hadronic radiation**
 2γ exchange
new structure
- purely weak corrections

Note: for NC-scattering, straightforward separation
IR divergences: need to combine real and virtual radiation

H. Spiesberger (Mainz)

MCEGs, 20. 2. 2019 5 / 20

Hubert Spiesberger (Mainz): QED corrections for electron scattering

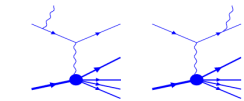
- High-precision measurements need careful treatment of radiative corrections.
- Closely related to experimental conditions need full Monte Carlo treatment (Unfolding) including simulation of hadronic final states.
- The basics are known and available ...
- ... but improvements are needed.

Radiative corrections in SIDIS

The Born cross section



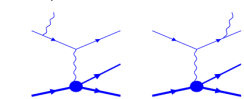
Emission of a radiated photon (semi-inclusive processes)



Loop diagrams



Emission of a radiated photon (exclusive processes)



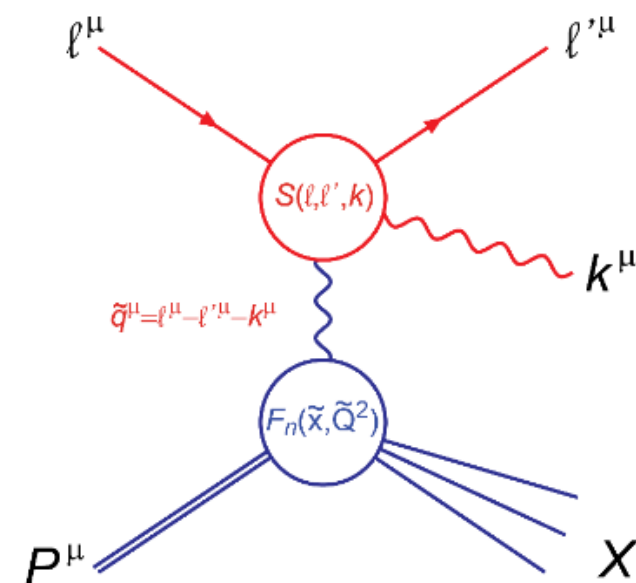
The real polar angle of virtual photon is changing due to radiation of the real photon, introducing azimuthal dependence, coupling to ϕ -dependence of the x-section
Akushevich, Ilyichev, Osipenko, PL B672 (2009) 35

Andrei Afanasev (GWU): Semi-analytic vs. Monte-Carlo Approaches for QED Corrections to SIDIS

- Consistent approach to address RC for SSA in polarized SIDIS
- SSA due to two-photon exchange need to be included in analysis of SSA from strong interaction, of same size at JLAB experiments
- More detailed calculation of the two-photon exchange at quark level required: elastic scattering, inclusive, semi-inclusive, and exclusive DIS

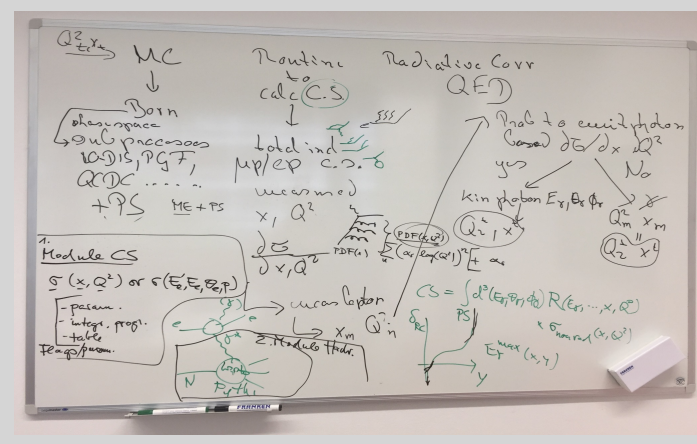
Radiative effects

- change kinematics on an event by event basis:
 - smearing of kinematic distributions
- change of virtual-photon direction:
 - false asymmetries in the azimuthal distribution of hadrons
- correction:
 - unfolding procedure, requires MCEG including radiative corrections / effects



ESC Radiative effects library

- Elke-Caroline Aschenauer, Andrea Bressan
- essential for high-precision measurements at the EIC
- collaboration with Hubert Spiesberger:
 - start back from HERACLES part of Djangoh
 - work on interface to PYTHIA6/8



MCEG2019: Next steps

- **General-purpose MCEGs**, HERWIG, PYTHIA, and SHERPA, will be significantly improved w.r.t. MCEGs at HERA time:
 - MCEG-data comparisons in Rivet will be critical to tune the MCEGs to DIS data and theory predictions.
 - The existing general-purpose MCEG should soon be able to simulate NC and CC unpolarized observables also for eA. A precise treatment of the nucleus and its breakup is needed.
 - First parton showers and hadronization models for ep with spin effects, but far more work needed for polarized ep / eA simulations.
 - Need to clarify the details about merging QED+QCD effects (in particular for eA).
- **TMD physics**
 - Vibrant community working on various computational tools for TMDs.
 - CASCADE: MCEG for unpolarized TMDs at high energy.
 - Need more verification of MCEG models with TMD theory / phenomenology.
- **GPD physics**
 - No modern MCEG available.
 - There is a path from PARTONS to a GPD MCEG, similar there is a project to extend MCEG for exclusive processes from JLAB12 to EIC.